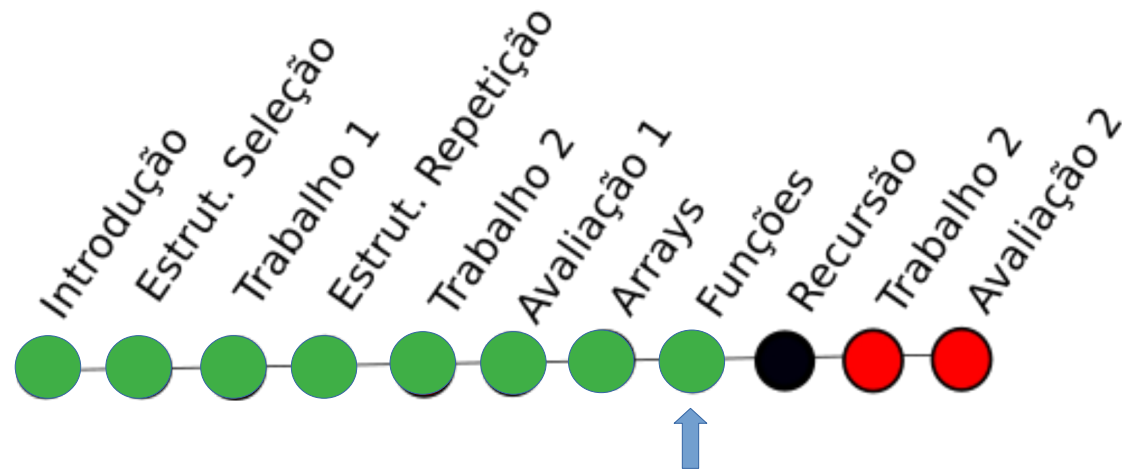


Tópico 08 - Funções

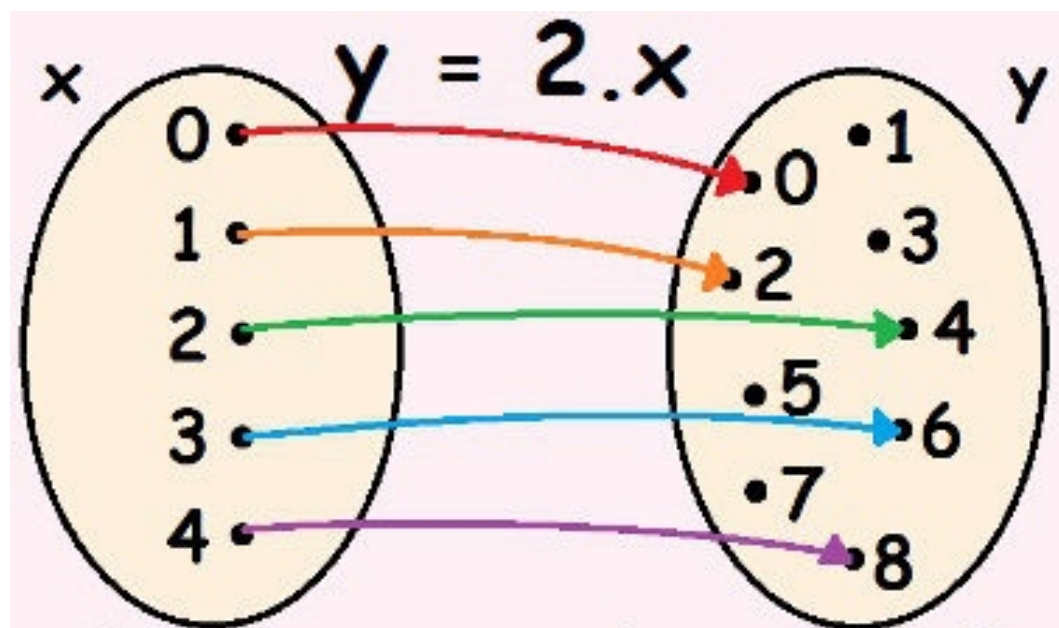
Prof. André Gustavo Hochuli

Plano de Aula

- Definição de Função
- Implementações de Funções
- Exercícios



Função

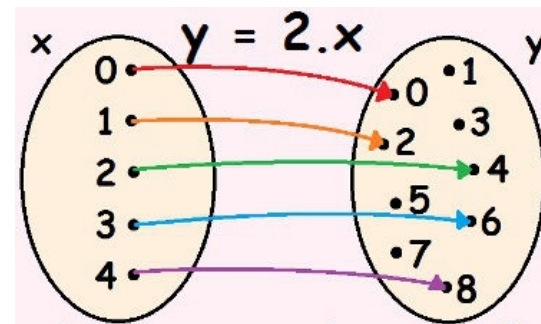


Função

- Bloco de código bem definido:

- Nome da Função
- Variáveis de Entrada
- Bloco de código
- Retorno

```
def f(x):  
    y = 2*x  
    return y
```



Função

- Fluxo

```
x = 10
```

```
y = f(x)
```

```
print(y) #20
```



```
def f(x):
```

```
    y = 2*x
```

```
    return y
```

```
x = 30
```

```
y = f(x)
```

```
print(y) #60
```



```
def f(x):
```

```
    y = 2*x
```

```
    return y
```

Função

- Contexto
 - O que ocorre na função é 'independente' do que ocorre em outra função. Ainda que tenham a mesma nomenclatura de variáveis
 - O resultado do que é computado na função, é retornado para quem a chamou pelo 'return'

```
def funcao_a():  
    y = 10  
    print('Função a: ',y)  
  
def funcao_b(y):  
    y = y*20  
    print('Função b: ',y)  
  
def funcao_c(x,y):  
  
    x *= 2  
    y *= -1  
    print('Função c: ',x,y)  
    return x+y
```

```
if __name__ == '__main__':  
  
    y = -60  
    x = 10  
    print(f'Main: {x}, {y}')  
  
    funcao_a()  
    print(f'Main: {x}, {y}')  
  
    funcao_b(y)  
    print(f'Main: {x}, {y}')  
  
    r = funcao_c(x,y)  
    print(f'retorno r: {r}')  
    print(f'Main: {x}, {y}')
```

Exercícios

- 1 - Crie uma função que imprima o triplo de um número
- 2 - Crie uma função que retorne a soma três variáveis
- 3 - Crie uma função que retorna a maior variável de um vetor
- 4 - Crie uma função que reverte um vetor

[40,10,-2,3] -> [3,-2,10,40]

Considerações Finais

- Funções ajudam na organização do código
- É muito útil quando um trecho de código precisa ser executado com distintas variáveis
- Auxiliam na reutilização de código
- Exercícios: [Tópico 08 – Funções](#)

Material de Apoio:

[Guanabara - Funções 1](#)

[Guanabara - Funções 2](#)